the haptic interface could be prevented from penetrating the virtual objects. The fiducial object does not penetrate the surfaces of the virtual objects. When the haptic interface does not penetrate the surface of a virtual object, the haptic interface and the fiducial object coincide. When the haptic interface penetrates the surface of the virtual object, the fiducial object remains located on the surface of the virtual object. The purpose of the fiducial object remaining on the surface is to provide a reference to the location on the surface of the virtual object where haptic interface would be if the haptic interface could be prevented from penetrating surfaces. It is important to know the location of the fiducial object in order to accurately determine the forces to be applied to the user. The method used to determine the fiducial object will be described in more detail below.

After the haptic rendering application determines both the haptic interface and the fiducial object, in step 18, the application calculates a force to be applied to the user in real space through the haptic interface device. After the haptic rendering application has calculated the force to be applied to the user, this force may be generated and applied to the user through a haptic interface device.

In the preferred embodiment of the method of the present invention, the haptic rendering application prevents the fiducial object from penetrating the surface of any of the virtual objects in the virtual environment. In this embodiment, the fiducial object is placed where the haptic interface would be if the haptic interface and the virtual object were infinitely stiff. Forcing the fiducial object to remain on the surface of the virtual object allows for a more realistic generation of the forces arising from interacting with the virtual object. Unlike in the vector field methods, the direction of the force to be applied to the user in real space is unambiguous. The user is not "pulled" through an object when the user should continue to be "pushed" away from the object. The method of the present invention is therefore suitable for thin objects and arbitrarily shaped polyhedral objects.

In yet another embodiment, the haptic rendering algorithm forces the fiducial object to follow the laws of physics in the virtual environment. This allows for an even more realistic simulation of the real world environment.

Referring again to Fig. 2, once the haptic rendering application has generated a representation of an object in graphic space, in step 22 the haptic interface device senses the position of the user in real space. In another embodiment, the haptic interface device senses the position of the user simultaneously with the haptic rendering application generating the representation of the object in graphic space. The haptic interface device may utilize any of the devices known in the art for sensing the position of an object.

After the haptic interface device has sensed the position of the user in real space, the information regarding the position of the user is relayed to the haptic rendering application. In step 24, the haptic rendering application uses the position of the user in real space to determine the location of the haptic interface point in graphic space. When the user changes position, the haptic interface device senses this change in position and the haptic rendering application updates the location of the haptic interface point in graphic space to reflect the change of the user's position in real space.

Once the haptic rendering application determines the haptic interface point location, it uses the haptic interface point location to determine the location of the fiducial object point in graphic space as illustrated by step 26. As discussed above, if the haptic interface point does not penetrate a virtual object, the haptic interface point and the fiducial object point are collocated. As the haptic interface point penetrates the surface of a virtual object, the fiducial object remains on the surface of the virtual object. The haptic rendering application computes the fiducial object point location to be a point on the currently contacted virtual object surface such that the distance of the fiducial object point from the haptic interface point is minimized. The method used by the haptic rendering application to calculate the location of the fiducial object will be discussed in more detail below.

In one embodiment, the location of the fiducial object point relative to the representation of the object is displayed on a display along with the representation of the object. When the position of the fiducial object changes, the display reflects this change in position.

The invention also relates to an apparatus for determining the forces to be applied to a user through a haptic interface. The apparatus includes a position sensor, a processor executing an algorithm to determine the forces to be applied to a user in real space, a display processor and a force actuator. In one embodiment, the algorithm determining the forces to be applied to the user includes a module generating a representation of an object in graphic space, a module determining the user's haptic interface in graphic space, a module determining the user's fiducial object in graphic space and a module calculating the force to be applied to the user in real space.

The present invention has the technical advantage of accurately replicating the touch sensations a user would experience when interacting with real world objects. The present invention has the further advantage of accurately modeling the forces applied to a user by thin and arbitrarily shaped polyhedral objects. The present invention has yet the further advantage of determining the appropriate forces to be applied to a user by a complex virtual object formed from overlapped simple virtual objects.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flowchart representation of an embodiment of a process for determining a force to be applied to a user through a haptic interface;

FIG. 2 is a flowchart representation of an embodiment of a process for determining a feedback force to be applied to a user through a haptic interface;

FIG. 3 is a pictorial view of a representation of a real world object in graphic space;

FIG. 4A is a pictorial view of a convex portion of a virtual object formed by two planar surfaces and a fiducial object located on one of the surfaces;

FIG. 4B is a pictorial view of the two planar surfaces of FIG. 4A and the fiducial object of FIG. 4A transitioning between the two planar surfaces;

FIG. 4C is a pictorial view of the two planar surfaces of FIG. 4A and the fiducial object of FIG. 4A after the fiducial object has transitioned between the surfaces;

FIG. 5A is a pictorial view of a concave portion of a virtual object formed by two planar surfaces and a fiducial object located on one of the surfaces;

FIG. 5B is a pictorial view of the two planar surfaces of FIG. 5A after the fiducial object has penetrated one of the surfaces;

FIG. 6A is a perspective view of a complex virtual object formed from two simpler virtual objects;

FIG. 6B is a cross-sectional view of the complex virtual object of FIG. 6A taken through line 6B—6B' of FIG. 6A;

FIG. 7 is a flowchart representation of an embodiment of a process for removing hidden surfaces of complex virtual objects;

FIG. 8 is a flowchart representation of an embodiment of a process for determining a friction force to be applied to a user through a haptic interface;

FIG. 9 is a graphical representation of a friction force applied to a user to model friction with slip;

FIG. 10 is a pictorial view of one of the triangular planar surfaces forming the surface of a virtual object;

FIG. 11 is a flowchart representation of an embodiment of a process for performing surface smoothing of a virtual object;

FIG. 12 is a flowchart representation of another embodiment of a process for performing surface smoothing of a virtual object;

FIG. 13 is a flowchart representation of embodiment of a process for modeling texture on the surface of a virtual object;

FIG. 14A is a pictorial view of one of the planar surfaces forming the surface of a virtual object;

FIG. 14B is a pictorial view of the texture map to be mapped onto the planar surface of FIG. 14A; and

FIG. 15 is a flow diagram of one embodiment of the invention.

Like reference characters in the respective drawn figures indicate corresponding parts.

## DETAILED DESCRIPTION OF THE INVENTION

In brief overview, and referring to FIG. 1, a flowchart shows the steps performed by one embodiment of the method of the present invention for determining the forces to be applied to a user through a haptic interface device. In step 10, the haptic rendering application generates a representation of a real world object in graphic space. As used herein, "rendering" is defined as the creation of an image in graphic space. "Haptic rendering application" refers to the application which generates the representation of the real world object and determines the forces to be applied to the user through the haptic interface. As used herein, "graphic space" is defined as the computer generated virtual environment with which the user can interact. In one embodiment, the haptic rendering application uses mathematical models to create the representation of the object. In another embodiment, a separate application is used to create the representation of the object. For example, in one embodiment, a Computer-aided design (CAD) software application is used to generate the representation of the object. The real world objects capable of being represented include planar surfaces, curved surfaces and arbitrarily shaped polyhedral objects. The real world objects may also include concave, convex and curved portions. As used herein, "virtual object" is defined as the representation of the real world object in graphic space.

In step 12, the sensors of the haptic interface system sense the position of the user in real space. As used herein, "real space" is defined as the real world environment. In step 14, the haptic rendering application utilizes the information obtained by the sensors to determine the haptic interface in graphic space. The location of the haptic interface describes the position of the user in the virtual environment. In step 16, the haptic rendering application determines the fiducial object in graphic space. The fiducial object is the "virtual" location of the haptic interface. The fiducial object location represents the location in graphic space at which the haptic interface would be located if the haptic interface could be prevented from penetrating the virtual objects. The fiducial object does not penetrate the surfaces of the virtual objects. When the haptic interface does not penetrate the surface of a virtual object, the haptic interface and the fiducial object coincide. When the haptic interface penetrates the surface of the virtual object, the fiducial object remains located on the surface of the virtual object. The purpose of the fiducial object remaining on the surface is to provide a reference to the location on the surface of the virtual object where haptic interface would be if the haptic interface could be prevented from penetrating surfaces. It is important to know the location of the fiducial object in order to accurately determine the forces to be applied to the user. The method used to determine the fiducial object will be described in more detail below.

After the haptic rendering application determines both the haptic interface and the fiducial object, in step 18, the application calculates a force to be applied to the user in real space through the haptic interface device. After the haptic rendering application has calculated the force to be applied to the user, this force may be generated and applied to the user through a haptic interface device.

In the preferred embodiment of the method of the present invention, the haptic rendering application prevents the fiducial object from penetrating the surface of any of the virtual objects in the virtual environment. In this embodiment, the fiducial object is placed where the haptic interface would be if the haptic interface and the virtual object were infinitely stiff. Forcing the fiducial object to remain on the surface of the virtual object allows for a more realistic generation of the forces arising from interacting with the virtual object. Unlike in the vector field methods, the direction of the force to be applied to the user in real space is unambiguous. The user is not "pulled" through an object when the user should continue to be "pushed" away from the object. The method of the present invention is therefore suitable for thin objects and arbitrarily shaped polyhedral objects.

In yet another embodiment, the haptic rendering algorithm forces the fiducial object to follow the laws of physics in the virtual environment. This allows for an even more realistic simulation of the real world environment.

In more detail and referring now to FIG. 2, a flowchart illustrates a more detailed sequence of steps performed by one embodiment of the present invention to determine a feedback force to be applied to a user in real space through a haptic interface. In the embodiment illustrated by the flowchart of FIG. 2, the user's interactions with the virtual environment are reduced to those of a point interacting with three dimensional objects. In other embodiments, the user's interactions are not reduced to those of a point interacting with three dimensional objects. In other embodiments, the haptic interface and the fiducial object may be a series of points. In still other embodiments, the haptic interface and fiducial object may be three-dimensional objects.

In step 20, the haptic rendering application generates a representation of a real world object in graphic space. As described above, this representation is termed the virtual object. The real world objects modeled by the method of the present invention may have concave portions as well as convex portions. Many different methods can be used to generate the virtual object. In one embodiment, the haptic rendering application defines the real world object as a mesh of planar surfaces. In one embodiment utilizing the mesh of planar surfaces method, each of the planar surfaces comprising the mesh has the same number of sides and the same number of nodes. In another embodiment, the planar surfaces comprising the mesh have varying numbers of sides and nodes. In the preferred embodiment, each of the planar surfaces is triangular and has three nodes. In another embodiment, the haptic rendering application defines the real world object as an n-noded polygon. In still another embodiment, the haptic rendering application describes the real world object using a coordinate system. In yet another embodiment, the representation of the object is displayed on a display.

FIG. 3 shows an example of a representation of a real world object which has been generated by one embodiment of the present invention. The real world object depicted in FIG. 3 is a space shuttle. The representation consists of 616 polygons. In one embodiment, the representation is generated from a standard object file format such as AutoCad's DXF or WAVEFRONT's OBJ.

Referring again to FIG. 2, once the haptic rendering application has generated a representation of an object in graphic space, in step 22 the haptic interface device senses the position of the user in real space. In another embodiment, the haptic interface device senses the position of the user simultaneously with the haptic rendering application generating the representation of the object in graphic space. The haptic interface device may utilize any of the devices known in the art for sensing the position of an object.

After the haptic interface device has sensed the position of the user in real space, the information regarding the position of the user is relayed to the haptic rendering application. In step 24, the haptic rendering application uses the position of the user in real space to determine the location of the haptic interface point in graphic space. When the user changes position, the haptic interface device senses this change in position and the haptic rendering application updates the location of the haptic interface point in graphic space to reflect the change of the user's position in real space.

Once the haptic rendering application determines the haptic interface point location, it uses the haptic interface point location to determine the location of the fiducial object point in graphic space as illustrated by step 26. As discussed above, if the haptic interface point does not penetrate a virtual object, the haptic interface point and the fiducial object point are collocated. As the haptic interface point penetrates the surface of a virtual object, the fiducial object remains on the surface of the virtual object. The haptic rendering application computes the fiducial object point location to be a point on the currently contacted virtual object surface such that the distance of the fiducial object point from the haptic interface point is minimized. The method used by the haptic rendering application to calculate the location of the fiducial object will be discussed in more detail below.

In one embodiment, the location of the fiducial object point relative to the representation of the object is displayed on a display along with the representation of the object. When the position of the fiducial object changes, the display reflects this change in position.

Once the haptic rendering application has determined the locations of the haptic interface point and the fiducial object point, in step 28 the haptic rendering application calculates the stiffness force component of the feedback force to be applied to a user in real space through the haptic interface. The stiffness force represents the force that would be applied to the user in the real world by a real world object due to the stiffness of the surface of the object. Simple impedance control techniques can be used to calculate the stiffness force to be applied. In one embodiment, the haptic rendering application uses Hooke's law to calculate the stiffness force as illustrated by equation (1) below, wherein k is the stiffness of the virtual object's surface.

$$F_{stiffness}=k(x_{fiducial\text{-}object}-x_{haptic\text{-}interface}) \tag{1}$$

In equation (1), $F_{stiffness}$ represents the stiffness force to be applied to the user through the haptic interface, $x_{fiducial\text{-}object}$ represents the position of the fiducial object in graphic space, $x_{haptic\text{-}interface}$ represents the position of the haptic interface in graphic space and k represents the stiffness of the virtual object's surface. As shown by equation (1), to calculate the stiffness force, the haptic rendering application first calculates the displacement between the fiducial object point location and the haptic interface point location, rep-